TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 499 – 513
© TÜBİTAK
doi:10.3906/elk-1909-31

A new hybrid genetic algorithm for protein structure prediction on the 2D triangular lattice

Nabil BOUMEDINE, **Sadek BOUROUBI**[*]
Department of Operations Research, Faculty of Mathematics,
University of Sciences and Technology Houari Boumediene, Algiers, Algeria

Received: 05.09.2019 • **Accepted/Published Online:** 19.09.2020 • **Final Version:** 30.03.2021

Abstract: The flawless functioning of the protein is essentially related to its three-dimensional structure. Therefore, predicting protein structure from its amino acid sequence is a fundamental problem that draws researchers' attention in many areas. The protein structure prediction problem (PSP) can be formulated as a combinatorial optimization problem based on simplified lattice models such as the hydrophobic-polar model. In this paper, we propose a new hybrid algorithm that combines three different known heuristic algorithms: the genetic algorithm, the tabu search strategy, and the local search algorithm to solve the PSP problem. Regarding the evaluation of the proposed approach, we present an experimental study, where we consider the quality of the product solution as the main assessment criterion. Furthermore, we compared the proposed algorithm with state-of-the-art algorithms using a selection of well-studied benchmark instances.

Key words: Protein structure prediction, 2D triangular lattice, HP model, genetic algorithm, local search algorithm, tabu search strategy, minimal energy conformation

1. Introduction

In molecular biology, the native structure of proteins is considered as the most important indicator that determines their biological role [1]. Predicting the native structure of proteins, called conformation, is a challenging problem in biology that has an immediate appeal to physicists, mathematicians, and computer scientists. This problem is called the protein structure prediction (PSP). A slight modification in the native structure of some particular proteins or an error throughout its folding causes many serious diseases, such as Alzheimer's and mad cow [1, 2]. Furthermore, predicting the tertiaries' structure of a protein from its primary structure information has many applications in understanding and treating these diseases [3]. Besides, the correct behavior of proteins depends essentially on its minimal energy conformation. Simplified models have been proposed to reduce the complexity of PSP, the most widely used one is the hydrophobic-polar model (HP model) [4]. In this model, the free energy of a valid conformation is inversely proportional to the number of hydrophobic nonlocal bonds of H-H type (topological H-H contacts) existing in this conformation. This type of contact occurs if two of the nonconsecutive hydrophobic monomers occupy adjacent points in the lattice [5]. Moreover, each occurrence of this type of contact reduces the global energy value by one unit [6]. PSP is an optimization problem where the goal is to find a confirmation c^* of the specific protein chain that minimizes the total induced energy $E(c)$ in all possible set of conformations C, i.e. $c^* = argmin\{E(c) \mid c \in C\}$ [7]. As we mentioned above, H-H contacts reduce the induced energy. Thus, finding the minimum energy conformation

[*]Correspondence: bouroubis@gmail.com

(optimal conformation) amounts to find a conformation that maximizes the number of H-H contacts [6]. As one may expect, solving this problem is very difficult due to the exponential number of possibilities when the size of the string is large. The PSP has been proved to be NP-complete, even for simplified lattice models [8, 9]. Due to the complexity of the PSP, a number of well-known heuristic optimization algorithms have been proposed to solve this problem for the HP model. In the two-dimensional square lattice, the first genetic algorithm was introduced by Unger and Moult [10], and then followed by other versions (see [11, 12] for details). Later on, an improved genetic algorithm was proposed in [13]. Shmygelska et al. used the ant colony optimization algorithm (ACO) in [7, 14]. Furthermore, an adaptation of memetic algorithms were suggested in [15, 16]. The particle swarm optimization algorithm (PSO) was applied in [17]. Jiang et al. proposed a hybrid approach that combines the tabu search and the genetic algorithm [18]. The Immune algorithms are introduced by Cutello et al. in [19, 20]. Recently, Islam et al. have proposed a clustered memetic algorithm with local heuristics in [21]. In addition, a number of metaheuristic algorithms have been used to solve the PSP problem in the two-dimensional triangular lattice. In [22], the authors suggested a new hybrid algorithm, called hybrid genetic algorithm (HGA). This latter enhances the performance of a classical GA implementation by reducing the encountered conformations throughout the generational process. Furthermore, the authors showed considerable quality improvements when compared to a simple genetic algorithm implementation SGA [22]. Later in [23], the authors proposed a new approach based on the tabu search algorithm using a generalized local move set to improve the landscape exploration and the quality of the produced solutions. Moreover, two approaches have been proposed in [24], including the elite-based reproduction strategy-genetic algorithm (ERS-GA), and a hybridization of hill-climbing and genetic algorithm (HHGA) that combine the ERS-GA with a hill-climbing algorithm. Recently, an extended particle swarm optimization method (EPSO) was applied to PSP in [25]. A new approach called IMOG was proposed in [26], which combines ions motion optimization algorithm (IMO) with a greedy algorithm, the authors showed that IMOG algorithm has good search ability and stability using benchmark data sets.

The rest of this paper is organized as follows: In Section 2, we present the 2D triangular lattice and the HP model used in this study. Furthermore, we present a 0–1 mathematical program with a detailed description of solutions encoding, the objective function, and constraints. We present a detailed description of the proposed hybrid algorithm in Section 3. Besides, in Section 4, we present the experimental study and the obtained results compared with state-of-the-art approaches. Finally, we give the main conclusions in the last section.

2. Hydrophobic-polar simplified model in a 2D triangular lattice

In the HP model, twenty amino acids are represented by two letters H and P, referring to their hydrophobicity characterization chosen among the two following options: hydrophobic (H) or polar (P) [4]. For any given protein sequence of n amino acids, the HP model consists of converting this latter into another sequence $s = (s_1, s_2, \dots, s_n)$ such that each element of the sequence $s_i \in \{H, P\}$, represents the hydrophobicity of the corresponding amino acids in the protein sequence. As we show in Figure 1, each node of the two-dimensional triangular lattice has six neighbors [27]. Hence, we use the symbols $1, 2, 3, 4, 5$ and 6 to encode the following movement directions on the two-dimensional triangular lattice: right-up, up, left-up, left-down, down, and right-down, respectively.

Figure 2 represents a feasible conformation in the 2D triangular lattice model for the protein sequence of 20 amino acids given in the HP model by HPHPPHHPHPPHPHHPPHPH. The green points represent

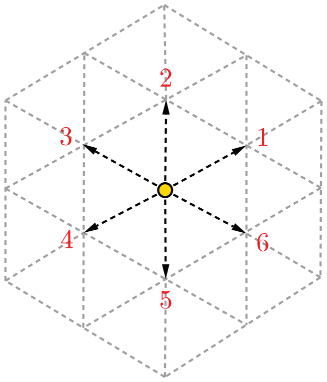


Figure 1. Six possible neighbors of a node in the 2D triangular lattice model.

the hydrophilic amino acids while the hydrophobic amino acids are represented in red. The energy of this conformation is $E(s) = -15$, i.e. 15 topological contacts of H-H type. We can represent a valid conformation by a sequence of $n-1$ movements in the lattice using the neighbor encoding given in Figure 1. For example, the movements sequence corresponding to the conformation given in Figure 2 is as follows: $mv(s) = [2,6,2,6,5,4,5,1,5,6,2,6,2,3,2,1,5,1,5]$. The sequence $mv(s)$ allows us to deduce the position of each amino acid in the lattice.

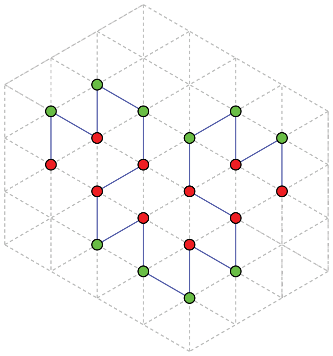


Figure 2. A feasible conformation in the 2D triangular lattice.

2.1. Mathematical program for the PSP problem

In this section, we suggest a mathematical program for the PSP problem in its 2D triangular lattice model. The aim here is to construct a mathematical program that can be implemented using mathematical modeling languages and compatible with the existing software solvers. Each node in a 2D triangular lattice has six neighbors represented in a lattice on a canonical basis. We consider the following encoding neighbors for a given position (i,j):

Let n be the length of the protein sequence, and let y_{ij}^k be a three-dimensional variable such that:

$$y_{ij}^k = \begin{cases} 1, & \text{if the position } (i,j) \text{ contain the } k^{th} \text{ amino acid in the protein sequence,} \\ 0, & \text{else.} \end{cases}$$

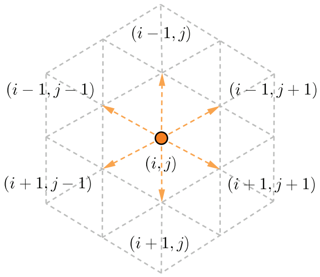


Figure 3. Six positions close to the position (i,j) in the 2D triangular lattice.

2.1.1. Constraints

First, we fix the first amino acid in the protein sequence at the position (n,n) as a starting point, i.e.

$$y_{nn}^{1} = 1.$$

Regarding the constraints, we can identify three different constraints that guarantee the admissibility of the resulting solution:

- A path in the lattice is a feasible solution if it occupies exactly n nodes in the lattice. This constraint can be written as follows:

$$\sum_{k=1}^{n}\sum_{i=1}^{2n}\sum_{j=1}^{2n} y_{ij}^{k} = n.$$

- A node in the lattice can contain at most one amino acid at the k^{th} position, hence:

$$\sum_{k=1}^{n} y_{ij}^{k} \leq 1,\ \forall i \in \{1,\dots,2n\},\ \forall j \in \{1,\dots,2n\}.$$

- A node in the lattice can contain the amino acid at the position $k+1$ if, and only if at least one of its neighboring nodes contains the k^{th} amino acid in the protein sequence:

$$y_{ij}^{k+1} \leq y_{i-1j+1}^{k} + y_{i-1j}^{k} + y_{i-1j-1}^{k} + y_{i+1j-1}^{k} + y_{i+1j}^{k} + y_{i+1j+1}^{k}, \forall i,j \in \{1,\dots,2n\},\ \forall k \in \{1,\dots,n-1\}.$$

2.1.2. Objective function

Let α_k be a numerical interpretation of any given amino acid into a binary value, where:

$$\alpha_k = \begin{cases} 1, & \text{if the } k^{th} \text{ amino acid in the protein sequence is hydrophobic, i.e. H,} \\ 0, & \text{if the } k^{th} \text{ amino acid in the protein sequence is hydrophilic , i.e. P.} \end{cases}$$

Thus, the objective function can be calculated as follows:

$$\max(\mathcal{Z}) = \frac{1}{2}\mathcal{Z}^{*} - \sum_{k=1}^{n-1} \alpha_k \alpha_{k+1},$$

where

$$\mathcal{Z}^* = \max_{y} \left\{ \sum_{i=1}^{2n} \sum_{j=1}^{2n} \left(\sum_{k=1}^{n} \alpha_k y_{ij}^k \right) \left(\sum_{k=1}^{n} \alpha_k \left(y_{i-1j+1}^k + y_{i-1j}^k + y_{i-1j-1}^k + y_{i+1j-1}^k + y_{i+1j}^k + y_{i+1j+1}^k \right) \right) \right\}.$$

We notice that this objective function is quadratic. In addition, this mathematical model guarantees an optimal solution which is included in the lattice enclosed by the points $\{(1,1),(1,2n),(2n,1),(2n,2n)\}$, with a starting point (n,n). The choice of these limit points is based on the fact that in a path graph P_n on the lattice, the maximal distance between the fixed starting node and the rest of the nodes (i.e. essentially for the last node) is at most equals to $n-1$ movements. More precisely it represents the radius of the graph, which is the number of edges in the case of a path graph P_n. With the view to its space complexity $O(n^3)$, this model has a rather high cost in terms of memory. However, it provides a rather good equilibrium with the computational complexity, since all variables are binary strings.

3. New hybrid approach for the PSP problem

The most commonly used hybridization in literature consists to combine two metaheuristics, one based on a single solution, known as s-metaheuristics, and the other based on a population, known as p-metaheuristics. The s-metaheuristics have proved their effectiveness for intensification, while the p-metaheuristics known by their exploration capacity [28]. Thus, this type of hybridization allows to establish a good balance between the diversification and the intensification of the research process [29, 30]. In this work, we propose a hybrid approach to solve the PSP problem, called GALSTS, referring to the adopted combination of the following metaheuristics: genetic algorithm, local search, tabu search strategy. The procedure of the proposed method is shown in algorithm 1. As all population-based approaches, the first step of GALSTS is to generate an initial population P of m feasible solutions (see Algorithm 2). The individuals of the new population P_{new}, are generated by tabu search crossover operator and improved by local search algorithm, in which the crossover operator is guided by a prohibition mechanism with a FIFO tabu list that prohibits the use of previously applied movements and to explore new regions in the search space. At each iteration the reproduction strategy of the suggested algorithm consists of selecting pairs of solutions (called parents) from the current population using the roulette wheel selection operator. Then, we generate iteratively a set of offsprings for the same selected parents by applying the tabu search crossover operator with random cut points. If a given crossover point is selected more than once, the crossover operation generate a solution that have been previously visited. In order to avoid this previously generated movements, we use a static tabu list T with a short-term memory, which is initially $T=\emptyset$. This list stores the k crossover points previously applied. At each iteration the tabu list T, is updated, the last crossover point is added in the list T and the oldest one is removed from T. All the points stored in T are excluded from the selection for the next iteration. This reproduction strategy is applied if a random generated $u \in [0,1]$ is less than a fixed probability p_c, otherwise, two offsprings will be generated by applying random mutation operator to the selected parents in order to maintain the diversity of the population in GALSTS (see Algorithm 3).

As to improve the quality of the offsprings produced by the crossover operator, each one of them is introduced with a probability p_m as an initial solution of a local search algorithm, such that the transition from a solution s to one of its neighbors s' is carried out by a random choice of amino acid i and replacing its direction by one of the other possible directions. If the quality of s' is better than s, then it replaces the current solution for the next iteration. This process is repeated until the satisfaction of the stopping criterion (see Algorithm 4).

This improvement phase allows to use of the information provided by parents more efficiently to produce high quality of solutions. The two solutions with the lowest fitness value are introduced into the new population P_{new}, if they do not already exist in the current population. The best m solutions of $P \cup P_{new}$ replace the individuals in the population P for the next generation. This approach is intended to avoid the premature convergence towards local optima, ensures a wide diversification of the current solutions throughout the search space explorations, and thus ensures the quality of solutions.

Algorithm 1 Suggested hybrid algorithm GALSTS.

Input: A protein sequence of n amino acids.

Output: The best confirmation for the protein sequence.

```
Begin
  Initialization: $P \leftarrow$ Generate a initial population of $m$ solutions by applying Algorithm 2;
  while the stoping criterion is not met do
    Create a new population by the following steps:
    $k \leftarrow 0,\ P_{new} \leftarrow \emptyset$;
    while $k \leq m$ do
      Select two parents $(p_1, p_2)$ from $P$ using the roulette wheel selection operator.;
      if $random\ u \in [0,1] < p_c$ then
        $offsprings \leftarrow$ Generate offsprings from the same parents by applying the crossover operator
        guided by the tabu search (see algorithm 3);
        Improve the quality of the offsprings using the local search algorithm (see algorithm 4);
      else
        $offsprings \leftarrow$ Mutate $(p_1, p_2)$;
      end if
      if $best_{offsprings}$ does not exist in the new population $P_{new}$ then
        $P_{new} \leftarrow P_{new} \cup \{best_{offsprings}\}$;
        $k \leftarrow k + 2$;
      end if
    end while
    $P \leftarrow$ the $m$ best solution of $P \cup P_{new}$;
  end while
End
```

The mechanisms of the suggested approach can be summarized by the following steps:

Step 1 Generate m solutions for the initial population using Algorithm 2.

Step 2 Evaluate the fitness of each solution.

Step 3 Create a new population by the following steps:

Step 3.1 Select two parents from the m solutions.

Step 3.2 Apply Algorithm 3, with a probability p_c.

Step 3.3 Apply Algorithm 4, with a probability p_m.

Step 3.4 Replace the previous population, with the current population.

Step 4 If the stopping criterion is not met go to Step 2.

3.1. Initial population

We start with a random initial population of m individuals. An initialization algorithm is proposed to allow generating only valid conformations for the initial population of GALSTS. We use a T list containing all the positions used in the current solution. We put the first amino acid at one point in the lattice and save their position in T. Then, for each amino acid, we select a random direction among the six directions given in Figure 1. If the selected direction generates an already occupied position existing in T, we randomly generate a new direction. If all six directions create an existing position in T, i.e. all directions create an invalid solution, we generate a new solution (see Algorithm 2).

Algorithm 2 Generation algorithm for the initial population.

Input: A chain length n.

Output: A feasible conformations for the protein sequence.

```
Begin
  Initialization: $T \leftarrow \emptyset$; // Table containing the position of each amino acid in the lattice.
  $T[1] \leftarrow (x_1, y_1)$; // Put the first amino acid in one node of the lattice.
  $i \leftarrow 2$;
  while $(i \leq n)$ do
      $K \leftarrow \{1, 2, 3, 5, 4, 6\}$; // The set of all possible directions on the 2D triangular lattice.
      $t \leftarrow true$;
      while $(t = true)$ do
          $u \leftarrow$ Random direction generated from the list $K$;
          $K \leftarrow K \backslash \{u\}$;
          $(x_i, y_i) \leftarrow$ The position generated by the direction $u$;
          if $(x_i, y_i) \notin T$ then
              $t \leftarrow false$;
              $T[i] \leftarrow (x_i, y_i)$;
          else if $(K = \emptyset)$ then
              $t \leftarrow false$;
              $i \leftarrow 2$;
          end if
      end while
      $i \leftarrow i + 1$;
  end while
End
```

3.2. Crossover operator

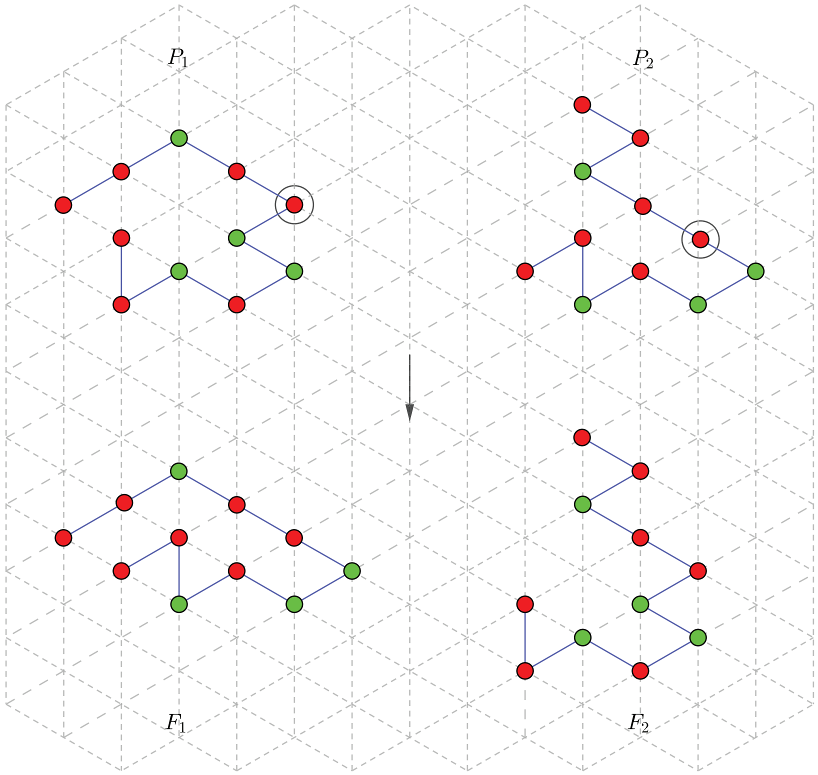
The crossover procedure consists of combining two or more solutions, called parents, to create other solutions, called offsprings. Many kinds of crossover methods exist. In our case, we chose a random cutting point operator, which consists to swap after selecting two parents p_1 and p_2, and generating a random cut point c_1, $1 < c_1 < n$, the parent subsequences limited by c_1 and n. As we show in Figure 4, the new two conformations (offsprings) f_1 and f_2, are obtained by combining p_1 and p_2 after generating a random cut point ($c_1 = 5$). The energy value of offspring f_1 is $E(f_1) = -7$, which is lower than the energy values of its parents, $E(p_1) = -2$ and $E(p_2) = -5$.

Algorithm 3 Reproduction algorithm guided by the local search algorithm and the tabu search strategy.

```
Input: Two selected parents $p_1, p_2$.
Output: Two offsprings produced by two parents.
Begin
  Initialization:
  $E_1 \leftarrow 0$, offspring$_1^* \leftarrow \emptyset$; $E_2 \leftarrow 0$, offspring$_2^* \leftarrow \emptyset$;
  $K \leftarrow 0$, $T \longleftarrow \emptyset$; // The tabu list.
  while the stopping criterion is not met do
      $K \leftarrow K + 1$;
      $u \leftarrow$ a random crossover point;
      if $u \notin T$ then
          (offspring$_1$, offspring$_2$) $\leftarrow$ crossover$(p_1, p_2)$;
          $u_1 \leftarrow$ random $[0, 1]$;
          if $u_1 \leq p_m$ then
              offspring$_1 \leftarrow$ local search(offspring$_1$);
              offspring$_2 \leftarrow$ local search(offspring$_2$);
          end if
          if $E$(offspring$_1$) $\leq E_1$ then
              offspring$_1^* \leftarrow$ offspring$_1$;
              $E_1 \leftarrow E$(offspring$_1$);
          end if
          if $E$(offspring$_2$) $\leq E_2$ then
              offspring$_2^* \leftarrow$ offspring$_2$;
              $E_2 \leftarrow E$(offspring$_2$);
          end if
      end if
          $T \longleftarrow T \cup \{u\}$;
  end while
  return (offspring$_1^*$, offspring$_2^*$);
End
```

Algorithm 4 Local search

```
Input: A feasible solution $s$ and its associated energy value $E(s)$.
Output: The best-found solution $s_{best}$.
Begin
  Initialization:
  $E(s_{best}) \leftarrow E(s)$;
  $s_{best} \leftarrow s$;
  while the stopping criterion is not met do
      $u \leftarrow$ a random mutation point ;
      $s \leftarrow$ applying the mutation operator over the solution $s$ at the position $u$;
      Evaluate $s$, i.e. calculate $E(s)$;
      if $E(s) < E(s_{best})$ then
          $s_{best} \leftarrow s$;
          $E(s_{best}) \leftarrow E(s)$;
      end if
  end while
End
```



The circled nodes indicate the cutting points positions.

Figure 4. Crossover operator applied on two conformations of the sequence $H^2PH^2P^2HPH^2$. The circled nodes indicate the cutting points positions.

3.3. Mutation operator

Generally, it consists of modifying some components, called genes, from an existing solution, to introduce more diversity into the solutions; they are generally applied with low probability p_m. In the proposed local search algorithm (see Algorithm 3), the neighbors of a given solution are defined similarly with the mutation operator, but the performed movement is acceptable if the quality of the current solution is improved. As we show in Figure 5, the new conformation s_m is obtained by applying the mutation operator on a solution s at the mutation red point 10, by changing the direction from 3 to 6. The energy value of s_m is $E(s_m) = -7$, which is less than the energy of s, $E(s) = -5$.

3.4. Selection operator

It is a technique that prefers the best solutions to participate in the reproduction phase, to create new solutions with satisfactory quality. In this work, we used the roulette wheel selection technique. This latter consists of associating for each solution i, between m solutions, a probability of selection P_i according to its fitness value f_i as follows:

$$P_i = \frac{f_i}{\sum_{j=1}^{m} f_j}.$$

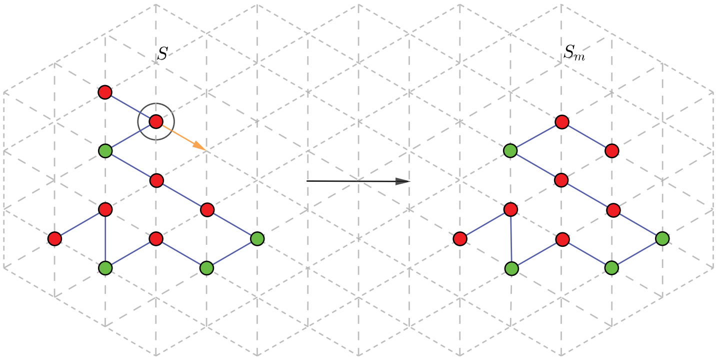


Figure 5. Mutation operator applied to the sequence $H^2PHP^2H^2PH^2$. The circled node indicates the position of the mutation point.

4. Experimental results and discussion

In this section, we present the conducted experimental study aiming to assess the performance of the suggested approach. For the following experimental study, we used several benchmark instances presented in the HP model of different sizes [10, 11]. Furthermore, we have selected the most used instances in the literature to conduct the forthcoming experiments. The used benchmark sequences are shown in Table 1, where the symbol $(\ldots)^i$ represents i fold repetitions of the respective subsequence in the brackets; for example, $(PH)^2$ is the simplified form of the sequence PHPH. In this experiment, the parameters we used for the suggested algorithm are fixed as follows: population size $m = 80$ for sequences 1 to 5, and $m = 120$ for sequences 6 to 10. For all sequences, the experiments ran the suggested algorithm for 100 generations. The tabu list size $k = 15$ for sequences 1 to 6, and $k = 20$ for the sequences 7 to 10. The parameter values p_c and p_m are fixed to 0.8 and 0.3, respectively.

Table 1. Used benchmark instances in the HP model.

Seq.	Length	Protein sequence in the HP model
1	20	$(HP)^2PH(HP)^2(PH)^2HP(PH)^2$
2	24	$H^2P^2(HP^2)^6H^2$
3	25	$P^2HP^2(H^2P^4)^3H^2$
4	36	$P(P^2H^2)^2P^5H^5(H^2P^2)^2P^2H(HP^2)^2$
5	40	$P^2H(P^2H^2)^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$
6	50	$H^2(PH)^3PH^4PH(P^3H)^2P^4(HP^3)^2HPH^4(PH)^3PH^2$
7	60	$P(PH^3)^2H^5P^3H^{10}PHP^3H^{12}P^4H^6PH^2PHP$
8	64	$H^{12}(PH)^2((P^2H^2)^2P^2H)^3(PH)^2H^{11}$
9	85	$H^4P^4H^{12}P^6(H^{12}P^3)^3HP^2(H^2P^2)^2HPH$
10	100	$P^3H^2P^2H^4P^2H^3(PH^2)^3H^2P^8H^6P^2H^6P^9HPH^2PH^{11}P^2H^3PH^2PHP^2HPH^3P^6H^3$

Table 2 shows a comparison of the best results taken from the above-stated algorithms used to solve the PSP problem in the 2D triangular lattice model, namely HGA [22], TS [23], ERS-GA [24], HHGA [24], IMOG [26], EPSO [25], and our suggested approach GALSTS. For each instance given in Table 2, we observe that the best solutions obtained by GALSTS are better or equal than all the cited approaches bellow. According to their energy values, all approaches can obtain the best known solution when the length of the sequence is less than

36. However, GALSTS is better than the other approaches for sequences 4, 6, 5, and 7. Moreover, the best solutions obtained by GALSTS are better than those of HGA, ERS-GA and SGA algorithms, for all sequences used in this experimental study. The number of possible solutions increases exponentially while the size of the instance increases. The best results obtained by GALSTS for the instance of large size demonstrates the ability of GALSTS to explore the search space more effectively compared with the other approaches. Instances larger than 64 are not covered in the literature for the 2D triangular lattice model. However, they were processed for the two-dimensional square lattice model [31]. According to the obtained results in Table 2, a strong improvement in energy appears compared with the 2D square lattice model.

Table 3, represented graphically in Figures 6 and 7, shows a performance comparison on the stability of our approach and three other algorithms HHGA, IMOG and ERS-GA, such that the efficiency of the algorithms is measured by the best and mean results in 30 independent runs for each sequence. We notice that for most instances, GALSTS can find the best optimal solutions and achieves better mean results than other algorithms. So the mean results obtained by GALSTS are very encouraging.

Table 2. The best conformations obtained by GALSTS compared with other algorithms.

Seq.	Length	SGA	HGA	TS	ERS-GA	HHGA	IMOG	EPSO	GALSTS	Conformation
1	20	–11	**–15**	**–15**	**–15**	**–15**	**–15**	NA	**–15**	Figure 8a
2	24	–10	–13	**–17**	–13	**–17**	**–17**	**–17**	**–17**	Figure 8b
3	25	–10	–10	**–12**	–12	**–12**	**–12**	**–12**	**–12**	Figure 8c
4	36	–16	–19	**–24**	–20	–23	**–24**	**–24**	**–24**	Figure 8d
5	48	–26	–32	–40	–32	–41	–40	–40	**–43**	Figure 8e
6	50	–21	–23	NA	–30	–38	**–40**	NA	**–40**	Figure 8f
7	60	–40	–46	**–70**	–55	–66	–67	NA	**–70**	Figure 8g
8	64	–33	–46	**–50**	–47	–63	–63	NA	**–67**	Figure 8h
9	85	NA	NA	NA	NA	NA	NA	NA	**–98**	Figure 8i
10	100	NA	NA	NA	NA	NA	NA	NA	**–87**	Figure 8j

Values in bold indicate the lowest energy for the correspondent instance. NA refers to not available data in the literature.

Table 3. A comparative study on the stability and the best prediction of GALSTS with other algorithms.

		ERS-GA		HHGA		IMOG		GALSTS	
Seq.	Length	Best	Mean	Best	Mean	Best	Mean	Best	Mean
1	20	**–15**	–12.50	**–15**	–14.73	**–15**	–14.73	**–15**	**–14.86**
2	24	–13	–10.20	**–17**	–14.93	**–17**	–14.93	**–17**	**–15.53**
3	25	12	–8.47	**–12**	–11.57	**–12**	–11.57	**–12**	**–12**
4	36	–20	–16.17	–23	–21.27	–23	–21.27	**–24**	**–21.93**
5	48	–32	–28.13	–41	–37.30	–41	–37.30	**–43**	**–39.86**
6	50	–30	–25.30	–38	–34.10	–38	–34.10	**–40**	**–37.6**
7	60	–55	–49.43	–66	–61.83	–66	–61.83	**–70**	**–68.26**
8	64	–47	–42.37	–63	–56.53	–63	–56.53	**–67**	**–58.46**
7	60	–55	–49.43	–66	–61.83	–66	–61.83	**–70**	**–68.26**
8	64	–47	–42.37	–63	–56.53	–63	–56.53	**–67**	**–58.46**

Values in bold indicate the best obtained evaluation for the correspondent instance.

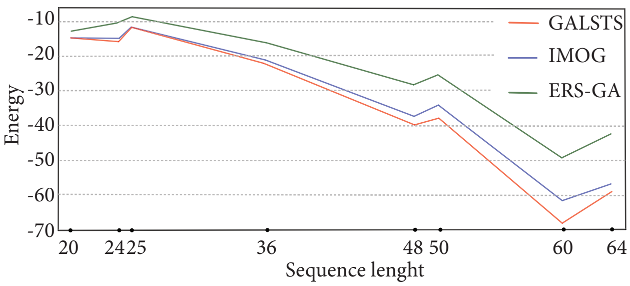


Figure 6. Mean energy comparison of GALSTS with IMOG and ERS-GA algorithms.

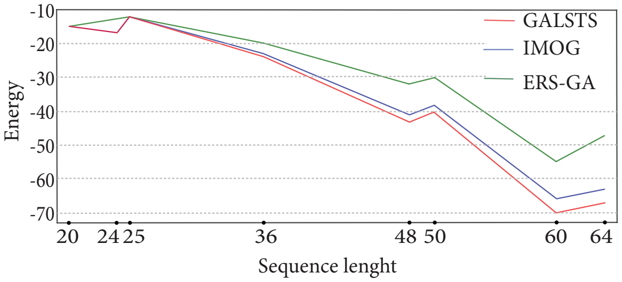


Figure 7. The best prediction comparison of GALSTS with IMOG and ERS-GA algorithms.

Table 4 summarizes the obtained results for 30 independent runs per each of the above-stated instances. This experiment aims to compare the suggested algorithm GALSTS with two competing algorithms ERS-GA and SGA. The results are derived according to the best and worst overall evaluation and their corresponding deviation from the best known value (BKV). The proposed algorithm GALSTS is proved to be more effective than the other competing algorithms, even when comparing its worst produced conformation to their best ones, with the sole exception of the first tested sequence, where it shows a slight difference. However, when increasing the size of the instance, it is clear that GALSTS is incrementally taking advantage over the competing algorithms, even when comparing its worst solution to their best ones. Furthermore, GALSTS can attain good quality conformations or even optimal, with the sole exception of the sixth tested instance, where it shows a one unit deviation of the best known evaluation.

Table 4. The best and worst evaluations comparison of GALSTS with SGA and ERS-GA algorithms.

			GALSTS		ERS-GA	SGA
Seq.	Length	E^*	Best (dev. BKV)	Worst	Best (dev. BKV)	Best (dev. BKV)
1	20	–15	**–15 (00)**	–14	**–15 (00)**	–11 (04)
2	24	–17	**–17 (00)**	–15	–13 (04)	–10 (07)
3	25	–12	**–12 (00)**	–12	**–12 (00)**	–10 (02)
4	36	–24	**–24 (00)**	–21	–20 (04)	–16 (08)
5	48	–43	**–43 (00)**	–38	–32 (11)	–26 (17)
6	50	–41	**–40 (01)**	–36	–30 (11)	–21 (20)
7	60	-	**–70** (-)	–65	–55 (-)	–40 (-)
8	64	-	**–67** (-)	–56	–47 (-)	–33 (-)

Values in bold indicate the best obtained evaluation for the correspondent instance and E^* is the best energy value.

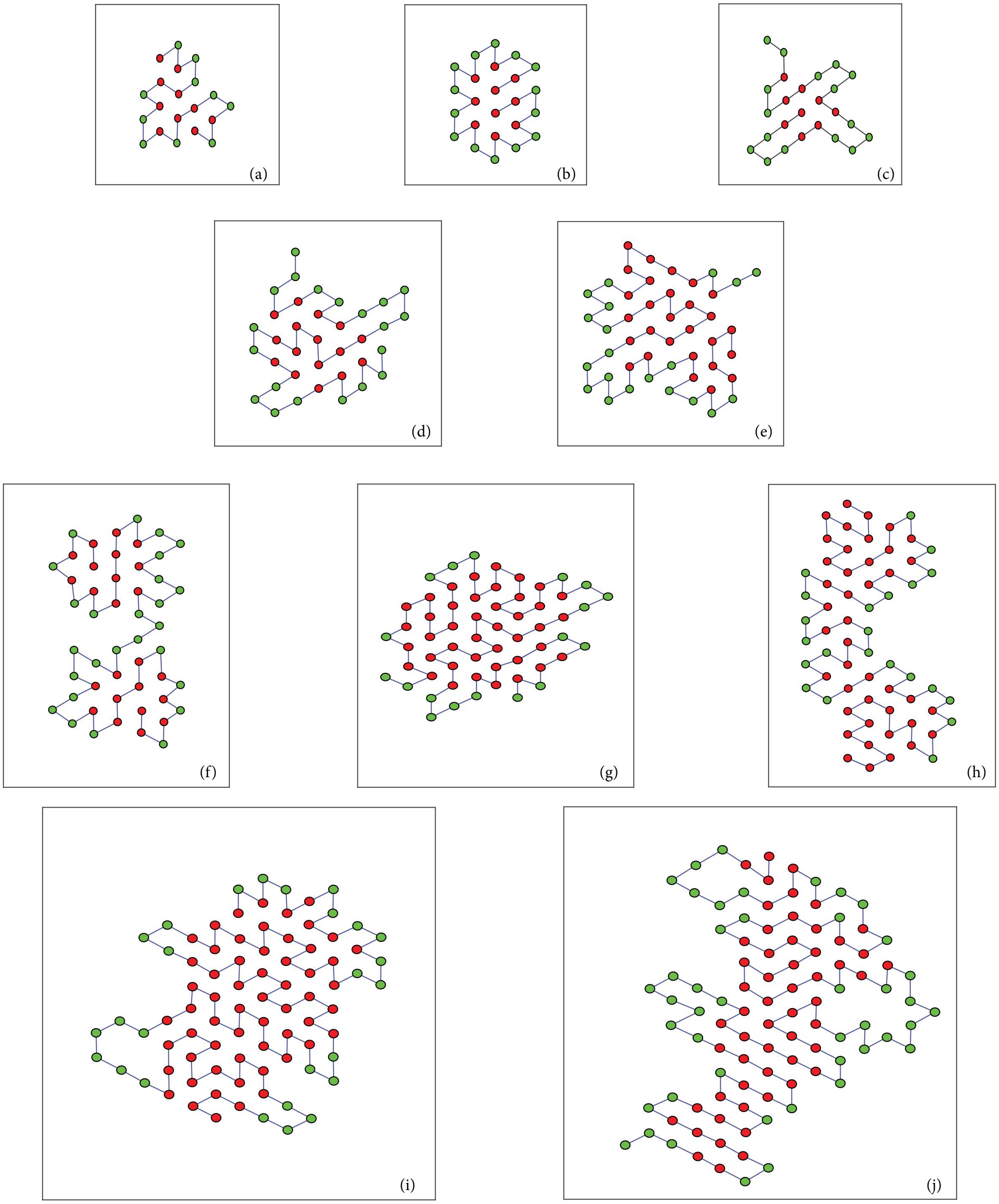


Figure 8. Results of the best conformation structure of ten protein sequences.

5. Conclusion

In this paper, we presented an efficient hybrid algorithm called GALSTS for solving the protein structure prediction in 2D triangular using the simplified hydrophobic-polar model. We suggested an initialization algorithm that allows generating only valid conformations for the initial population of GALSTS. This algorithm eliminates the reverse movements during the construction of solutions. GALSTS consists of using Tabu and Local Search algorithm to explore the search space more efficiently. From our experimental results, GALSTS can find the best known solutions and it is more effective than other existing algorithms in terms of stability. In terms of future scope of applications, GALSTS can be used to solve the PSP problem in the 3D cubic and 3D triangular lattice models; it can also be used to solve other optimization problems in the combinatorial optimization framework.

Acknowledgment

The authors thank N. Kantour and Y. Kherbachi for their generous help.

References

[1] Valastyan JS, Lindquist S. Mechanisms of protein-folding diseases at a glance. Disease Models & Mechanisms 2014; 7 (1): 9-14. doi: 10.1242/dmm.013474

[2] Hardy JA, Higgins GA. Alzheimer's disease: the amyloid cascade hypothesis. Science 1992; 256 (5054): 184-185. doi: 10.1126/science.1566067

[3] Chen Y, Ding F, Nie H, Serohijos AW, Sharma S et al. Protein folding: then and now. Archives of Biochemistry and Biophysics 2008; 469 (1): 4-19. doi: 10.1016/j.abb.2007.05.014

[4] Dill KA. Theory for the folding and stability of globular proteins. Biochemistry 1985; 24 (6): 1501-1509. doi: 10.1021/bi00327a032.

[5] Lau KF, Dill KA. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules 1989; 22 (10): 3986-3997.

[6] Lin CJ, Hsieh MH. An efficient hybrid Taguchi-genetic algorithm for protein folding simulation. Expert Systems with Applications 2009; 36 (10): 12446-12453.

[7] Shmygelska A, Hoos HH. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. BMC Bioinformatics 2005; 6 (1): 30.

[8] Berger B, Leighton T. Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete. In: Proceedings of the Second Annual International Conference on Computational Molecular Biology; Manhattan, NY, USA; 1998. pp. 30-39.

[9] Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M. On the complexity of protein folding. Journal of Computational Biology 1998; 5 (3): 423-465.

[10] Unger R, Moult J. Genetic algorithms for protein folding simulations. Journal of Molecular Biology 1993; 231 (1): 75-81.

[11] Dandekar T, Argos P. Folding the main chain of small proteins with the genetic algorithm. Journal of Molecular Biology 1994; 236 (3): 844-861.

[12] Hart WE, Krasnogor N, Pelta DA, Smith J. Protein structure prediction with evolutionary algorithms. Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation 1999; 39: 1596-1601.

[13] König R, Dandekar T. Improving genetic algorithms for protein folding simulations by systematic crossover. BioSystems 1999; 50(1): 17-25.

[14] Shmygelska A, Aguirre-Hernandez R, Hoos HH. An ant colony optimization algorithm for the 2D HP protein folding problem. In: International Workshop on Ant Algorithms; Brussels, Belgium; 2002. pp. 40-52.

[15] Krasnogor N, Blackburne BP, Burke EK, Hirst JD. Multimeme algorithms for protein structure prediction. In: International Conference on Parallel Problem Solving from Nature; Granada, Spain; 2002. pp. 769-778.

[16] Pelta DA, Krasnogor N. Multimeme algorithms using fuzzy logic based memes for protein structure prediction. In: Hart WE, Krasnogor N, Smith J (editors). Recent Advances in Memetic Algorithms. Berlin, Germany: Springer, 2005, pp. 49-64.

[17] Bautu A, Luchian H. Protein structure prediction in lattice models with particle swarm optimization. In: International Conference on Swarm Intelligence; Brussels, Belgium; 2010. pp. 512-519.

[18] Jiang T, Cui Q, Shi G, Ma S. Protein folding simulations of the hydrophobic–hydrophilic model by combining tabu search with genetic algorithms. The Journal of Chemical Physics 2003; 119 (8): 4592-4596.

[19] Cutello V, Morelli G, Nicosia G, Pavone M. Immune algorithms with aging operators for the string folding problem and the protein folding problem. In: European Conference on Evolutionary Computation in Combinatorial Optimization; Lausanne, Switzerland; 2005. pp. 80-90.

[20] Cutello V, Nicosia G, Pavone M, Timmis J. An immune algorithm for protein structure prediction on lattice models. IEEE Transactions on Evolutionary Computation 2007; 11 (1): 101-117.

[21] Islam MK, Chetty M. Clustered memetic algorithm with local heuristics for ab initio protein structure prediction. IEEE Transactions on Evolutionary Computation 2012; 17 (4): 558-576.

[22] Hoque MT, Chetty M, Dooley LS. A hybrid genetic algorithm for 2D FCC hydrophobic-hydrophilic lattice model to predict protein folding. In: Australasian Joint Conference on Artificial Intelligence; Hobart, Australia; 2006. pp. 867-876.

[23] Böckenhauer HJ, Ullah AZ, Kapsokalivas L, Steinhöfel K. A local move set for protein folding in triangular lattice models. In: International Workshop on Algorithms in Bioinformatics; Karlsruhe, Germany; 2008. pp. 369-381.

[24] Su SC, Lin CJ, Ting CK. An efficient hybrid of hill-climbing and genetic algorithm for 2D triangular protein structure prediction. In: 2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW); Hong Kong, China; 2010. pp. 51-56.

[25] Guo Y, Wu Z, Wang Y, Wang Y. Extended particle swarm optimisation method for folding protein on triangular lattice. IET Systems Biology 2016; 10 (1): 30-33.

[26] Yang CH, Wu KC, Lin YS, Chuang LY, Chang HW. Protein folding prediction in the HP model using ions motion optimization with a greedy algorithm. BioData Mining 2018; 11 (1): 17.

[27] Gillespie J, Mayne M, Jiang M. RNA folding on the 3D triangular lattice. BMC Bioinformatics 2009; 10 (1): 369.

[28] Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR) 2003; 35 (3): 268-308.

[29] Talbi EG. Metaheuristics: From Design to Implementation. Hoboken, NJ, USA: John Wiley & Sons, 2009.

[30] Raidl GR, Puchinger J, Blum C. Metaheuristic hybrids. In: Gendreau M, Potvin JY (editors). Handbook of Metaheuristics. Boston, MA, USA: Springer, 2010, pp. 469-496.

[31] Zhao X. Advances on protein folding simulations based on the lattice HP models with natural computing. Applied Soft Computing 2008; 8 (2): 1029-1040.